G4beamline

https://www.muonsinc.com/Website1/G4beamline

G4beamline utilise les mêmes bases que le code Geant 4 mais propose une interface utilisateur ne nécessitant pas une grande maitrise du langage C.

G4beamline

Getting Started | Documentation | Support | Download

G4beamline 3.08 is now available for Windows 10 and Windows 11.

G4beamline stands out from the many other beam simulation tools in two major ways:

- **it is considerably easier to use than most, and**
- **it performs a realistic simulation, including the decays and interactions of beam particles.**

G4beamline is capable of much more realistic simulations than most beam codes, and is used by over 500 researchers worldwide. G4beamline is a particle tracking simulation program based on Geant4; it is optimized for simulating beamlines, and is especially useful for muon facilities in which decays and/or interactions with matter are important. It can also simulate systems not organized as a beamline, and has both a cosmic-ray "beam" and an isotope decay "beam". G4beamline is an open-source program distributed for Linux, Mac OS X, and Windows; it can also be built from source.

G4beamline


g4blgui
File View Tools Help
Input file:
Browse
Directory:
Parameters:
Visualization (OpenGLStoredQt)
Events per Image: 1
Show Output
Run
Muons, Inc.
Innovation in Research
G4beamline
3.06
http://g4beamline.muonsinc.com
Introduction
Using the Program
Visualizing the System
HistoRoot - program to Create Histograms
References
Appendix: G4beamline Commands
Appendix: List of Particles (QGSP)
Downloading Geant4 Data Sets
NOTE:Before G4beamline can run, it requires that various Geant4 data sets be downloaded and unpacked; which specific data-sets are required depends on the physics list used in the input.file. When first run, G4beamline prompts the user to download them. If you find you need to download more data-sets, click here.
Introduction
G4beamline is a particle-at-a-time simulation program based on Geant4 [1] and optimized for the simulation of beamlines. See the G4beamline User's Guide at the above URL for a description of the program (it is also installed with this distribution in the doc directory).
This brief description only discusses how to use the program, it does not describe how to construct a simulation's input file;

G4beamline is designed to be very user friendly:

- No C++ programming is required to use it.
- The system and simulation are described in a single ASCII file
- The system description uses a self-describing language which can be easily understood by anyone familiar with beamline simulations.
- Advanced visualization capabilities are available out-of-the-box.
- Plots and histograms are easily generated using G4beamline commands to write NTuples, and external programs to plot them, such as AllPlot, HistoRoot, Root, and Gnuplot.
- All common beamline elements are implemented, such as dipole and quadrupole magnets, RF cavities, absorbers and targets, etc.
- Complete documentation for G4beamline is available in the G4beamline Users Guide.

http://www.muonsinternal.com/muons3/g4beamline/G4beamlineUsersGuide.pdf

Manuel et autres documents sur serveur des salles E104-E105 du bâtiment E
Lien : /home/public/G4BL

G4beamline (Pour G4beamline graphic users interface)

L'utilisation du code se fera via linux. (commande linux) cd, cd.. ,ls , emacs, more, mv,…… Un rappel des principales commandes est disponible dans dossier G4BL (fichier.pdf)

Important et Indispensable. Au démarrage ouvrir plusieurs fenêtres cahque fenêtre sera dédiée à une fonction bien précise. Ne pas oublier de lancer à partir du dossier de travail ou sont les fichiers .g4bl les trois commandes ci-dessous pour s 'attacher les 3 fichiers sources Géant4, root et G4beamline :

```
source /opt/cern/geant/geant4.10.05.p01/bin/geant4.sh
source /opt/cern/root/root_v6.22/bin/thisroot.sh
Et source /opt/G4beamline/G4beamline-3.06/bin/g4bl-setup.sh  (génération fichier root)
ou (suivant utilisation)
source /opt/G4beamline/G4beamline-3.08/bin/g4bl-setup.sh    (géométrie)
```

Exple avec fichier csi-cube-ref.g4bl

Visualisation Géométrie : g4blgui cf ppt4

Acquisition des données: commande: g4bl csi-cube-ref.g4bl

Génère un fichier g4beamline.root
à ouvrir uniquement sous root ou par python (uproot)

Visualisation des données: avec root cf ppt 7

Démarrage root : TBrowser b; (une fenêtre doit s'ouvrir)
sinon selectionner demo
.q pour quitter

Etape 1

g4blgui :

Récupérer dans la zone publique dans dossier: public/angelique

1 - Fichier g4bl.txt où sont écrites des instructions dont les commandes source indispensables .

2 - Récupérer dans votre domaine le fichier csi-cube.g4bl et csi-cube.root et exemples

Il servira de programme de base et vous permet de vérifier si tout fonctionne correctement.

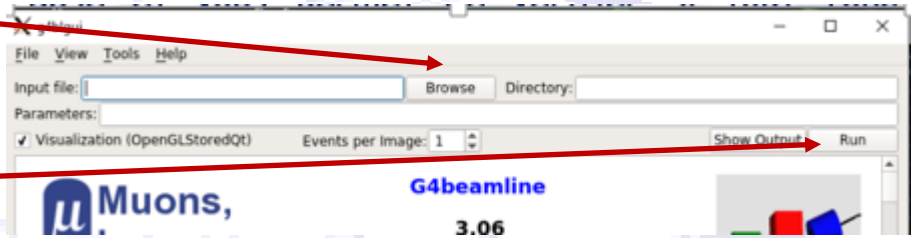
Passer la commande g4blgui

Une fenêtre identique à celle-ci-contre doit apparaître

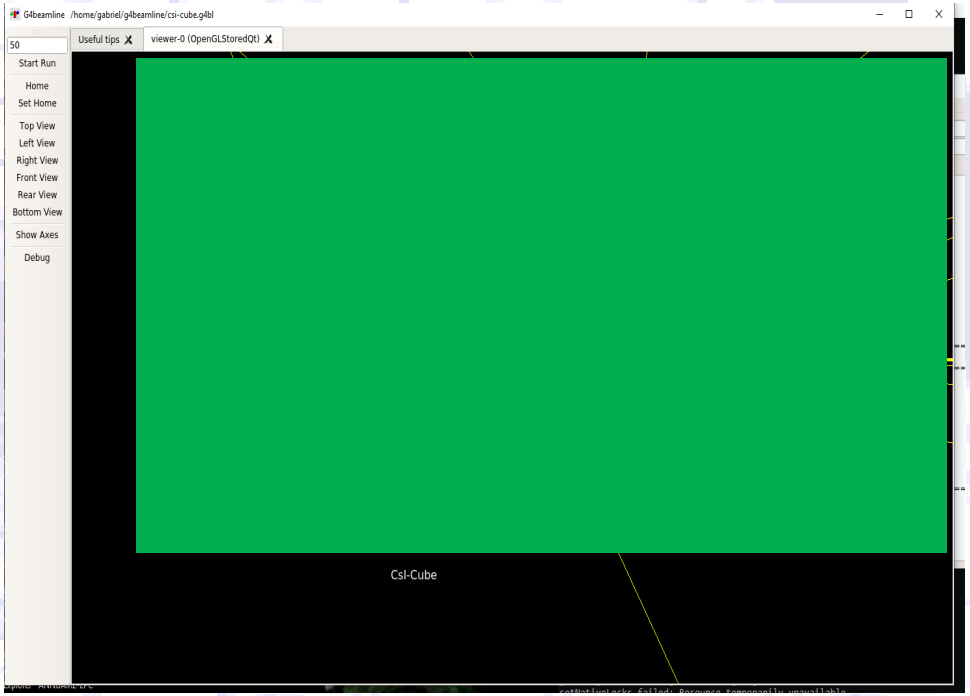
Etape 2

Avec Browse (en haut au centre) aller chercher le fichier csi-cube.g4bl

Actionner Run (en haut à droite)



Une fenêtre identique à celle-ci-contre doit apparaître
Elle permet de visualiser la géométrie



Par clic droit de la souris on peut sélectionner une fonction: zoom in , zoom out…et ainsi zoomer et faire tourner la géométrie.

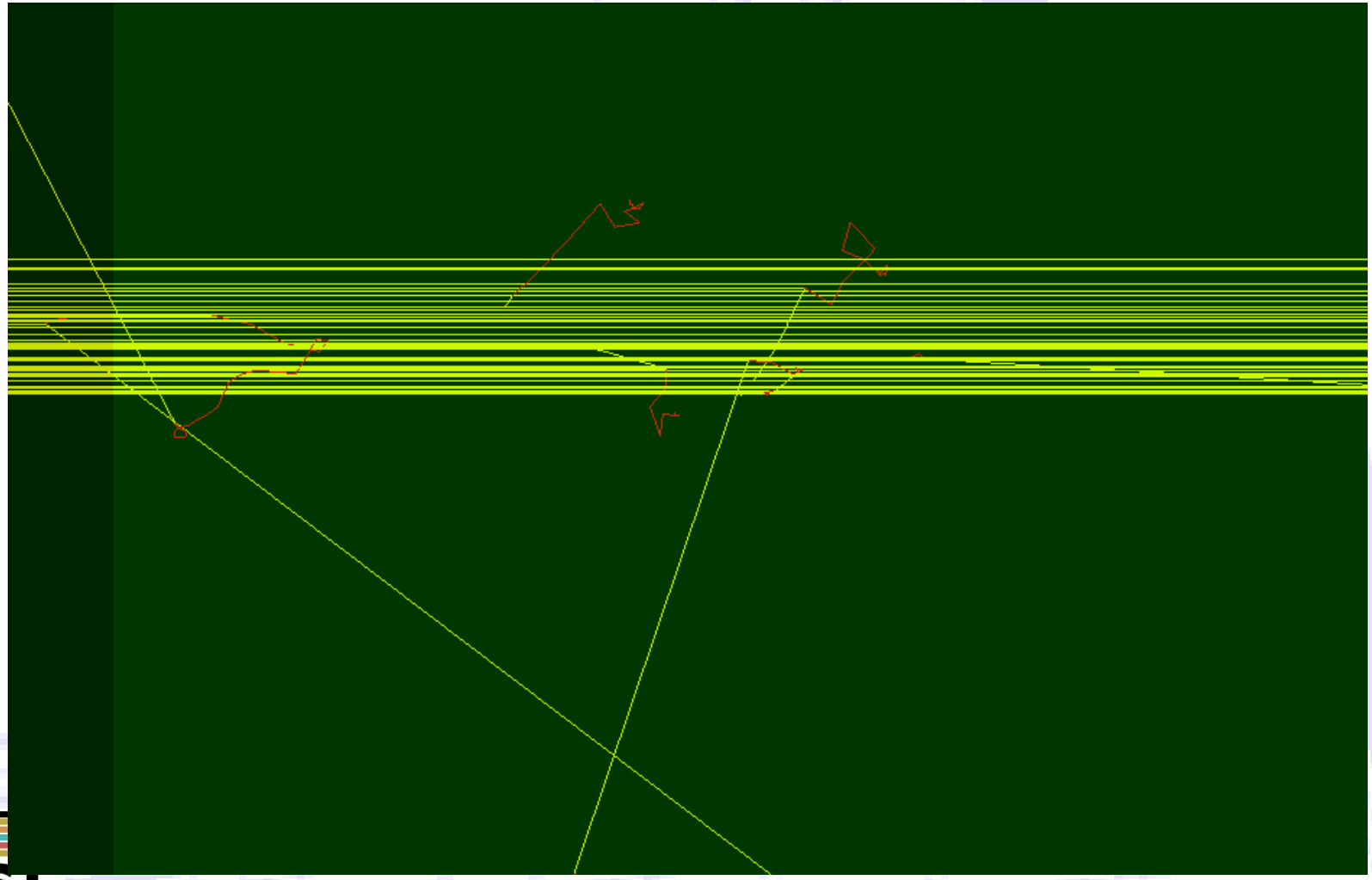
Apprenez a bien maitriser le maniement.

Etape 3

En sélectionnant un nombre de particules tests (exple ici 50) et en cliquant sur Start Run on voit apparaitre la trajectoire des 50 particules créées et les interactions de ces particules dans le détecteur de Csi.

Etape 3b

On peut zoomer pour voir plus en détail sur les interactions. Ainsi on peut voir la diffusion compton ou l'effet photoélectrique des photons (jaune) sur les électrons (rouge) du milieu.



Visualisation des spectres :

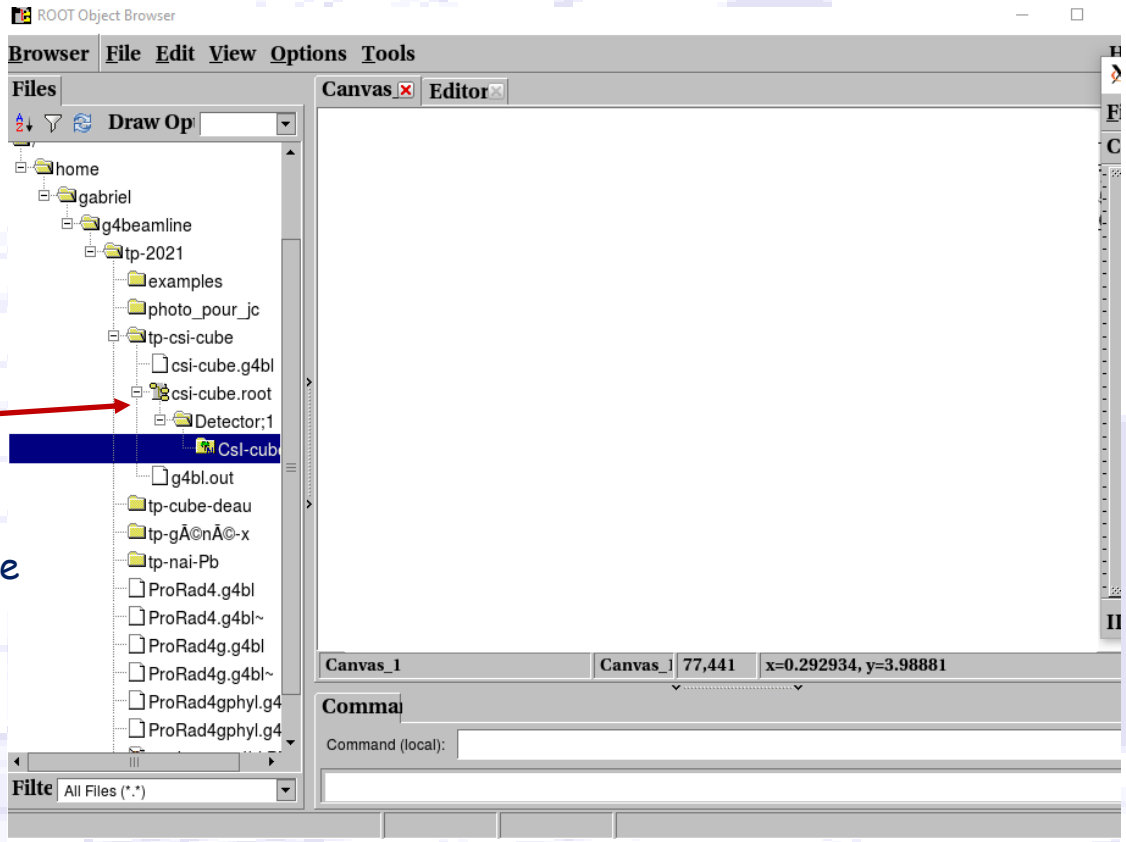
Sous Root :

Sur une deuxieme fenêtre de terminal executer la commande root
sous root enter la commande TBrowser b;

Une fenêtre identique à celle-ci-contre doit apparaître
Elle permet d'utiliser le logiciel d'analyse root.

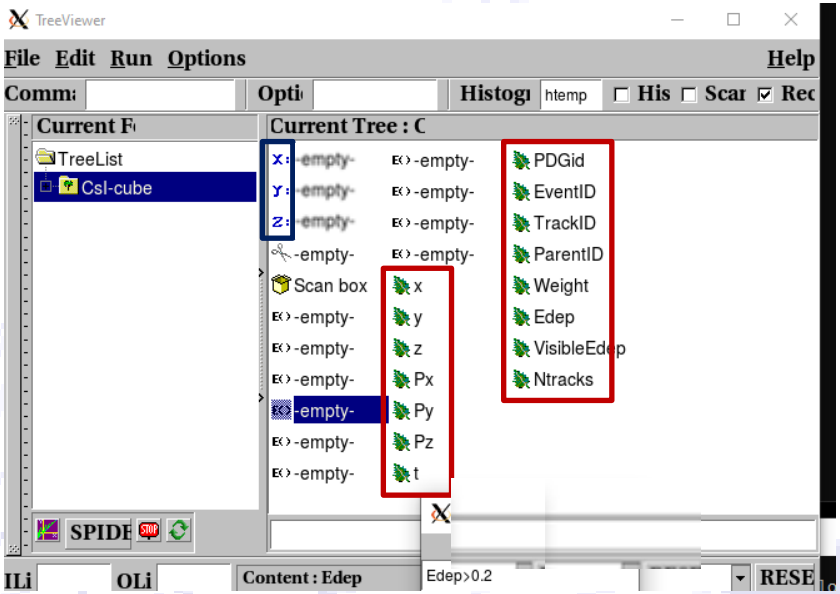
Sélectionner dans l'arborescence le fichier csi-cube.root

Double cliquer, la ligne detector doit apparaitre.



Etape 4b

Clic droit sur detector
et sélectionner
StartViewer sur menu déroulant.
Une fenêtre identique à celle-ci contre doit apparaître:



Vous avez l'ensemble des datas enregistrès lors de la simulation qui a générer le g4beamline.root renommé ensuite csi-cube.root (fichier.root) symboliser par les feuilles x, y, z,……jusqu'à Ntracks
PDGid Indentification des particules créées (gamma code 22, électron code 11,…)

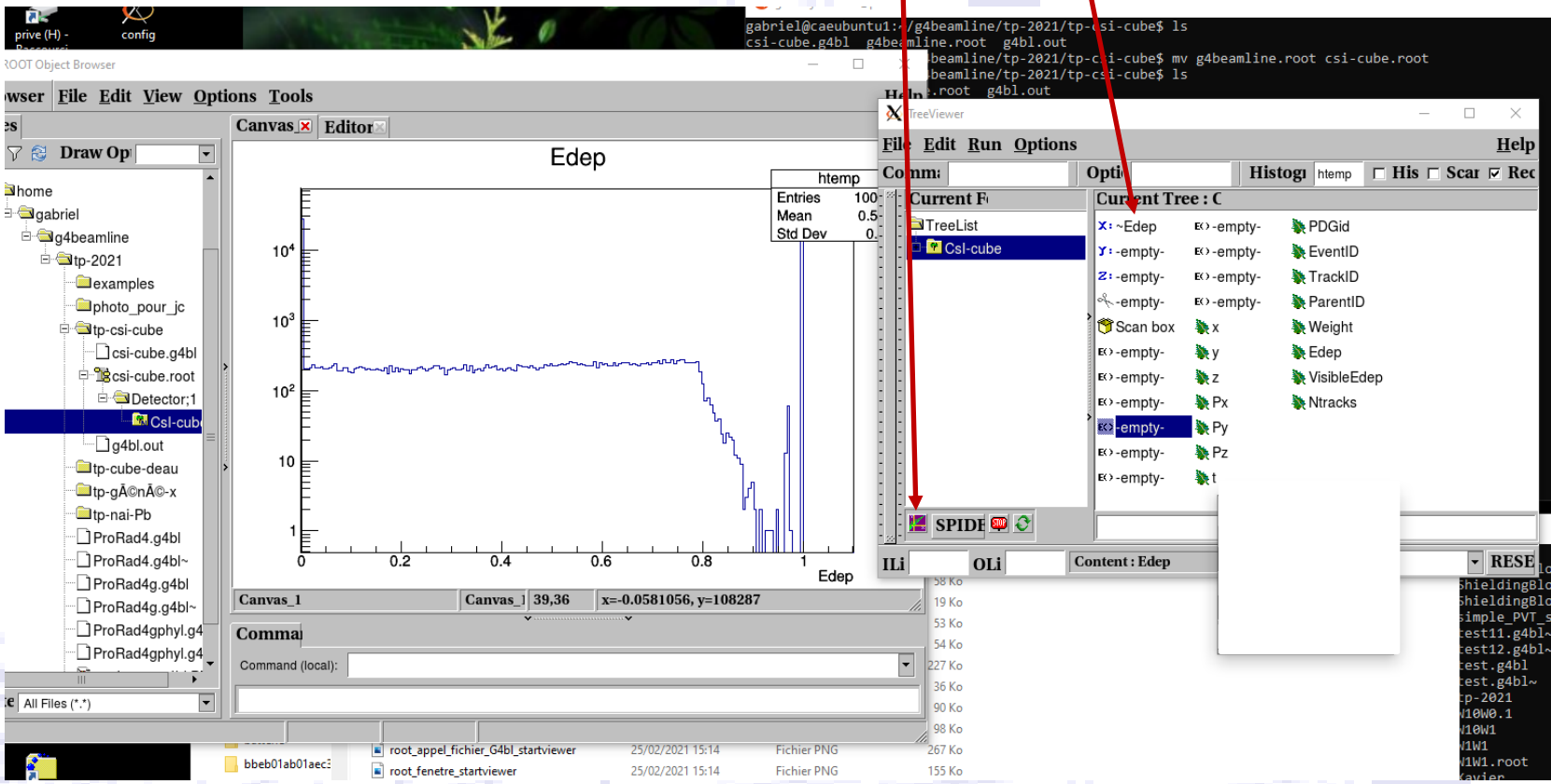
Nomenclature des Code des Particules: PDGid
https://twiki.cern.ch/twiki/pub/Geant4/ExtendingFnalDb/allParticles.txt

Edep : énergie déposée dans le détecteur

Pour tracer les spectres il faut remplir selon le choix les variables x, y ou/et z initialement vide (empty).

Etape 5

Exple: ici on a positionné (déplacer avec la souris la variable Edep dans la variable x
On visualise le spectre en cliquant sur le logo spectre (ne pas cliquer sur SPIDE)
Rmq Clic droit pour remettre empty la variable désirée

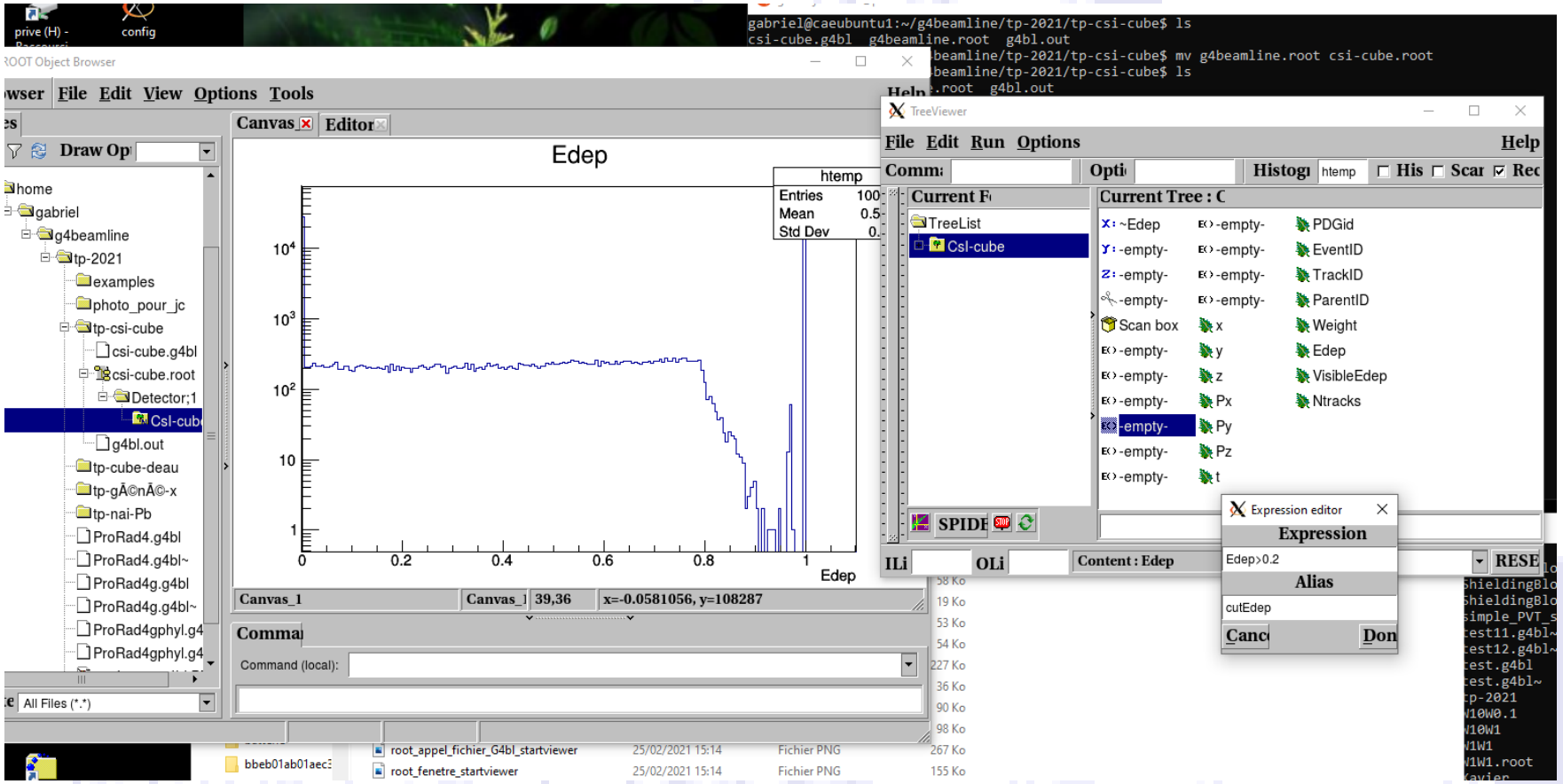


Rmq On a aussi par un clic droit sur l'axe des y mis en échelle logarithme y

On peut en déplaçant la souris connaître la valeur en X (Edep) et Y (nbre de coups) Dans le spectre. (binx=Edep binc=contenu en y)

Etape 6

On peut faire des opérations de coupure et de sélectionner des conditions sur les spectres.



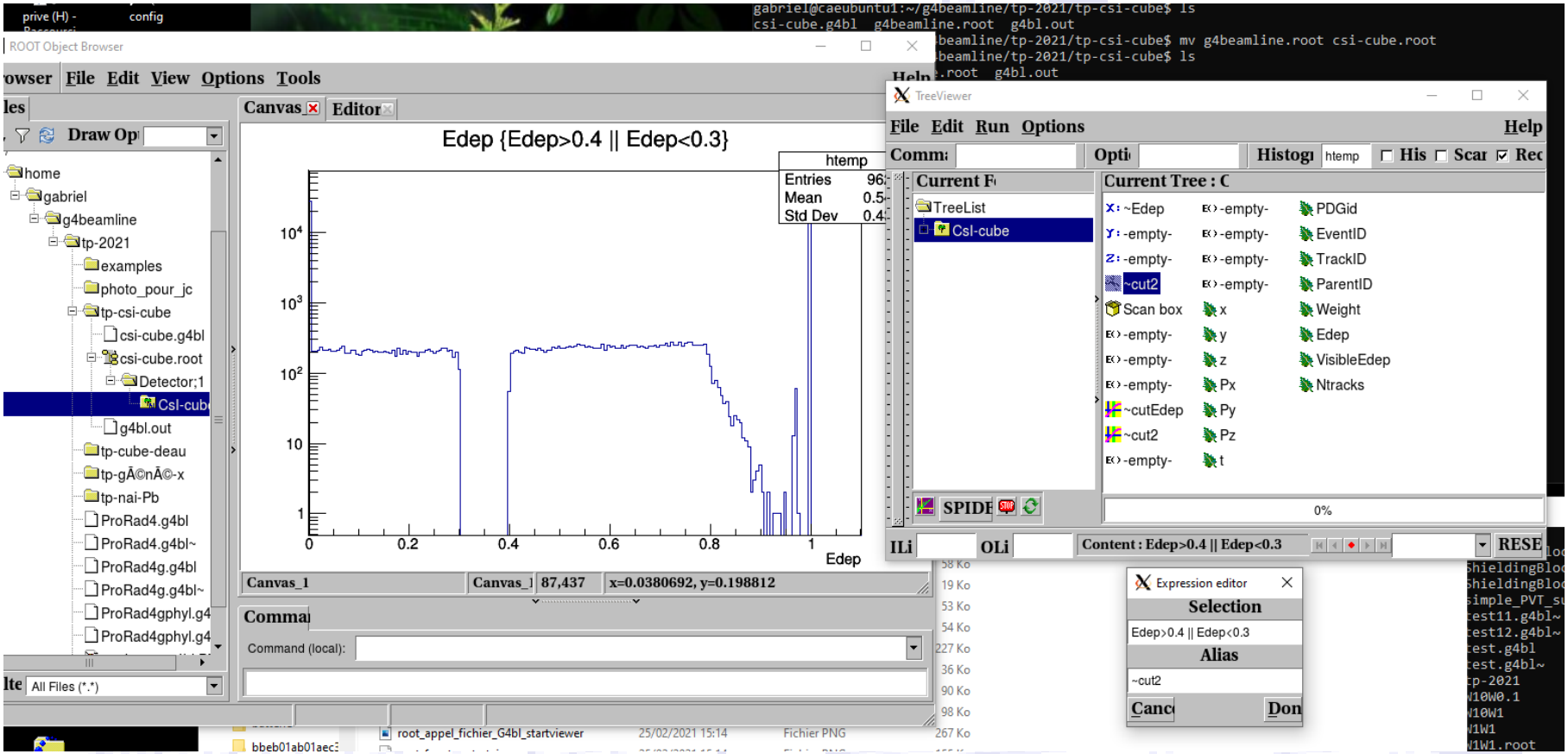
Ici on créer une sélection pour les énergies déposées supérieures à 0,2 MeV. La copuure sera nommée avec l'alias cutEdep

Etape 6b

Voici le spectre résultant
Coupure à 0,2 MeV

A noter qu on a déplacer l'alias cutEdep dans la selection indiquée par le symbole des ciseaux pour appliquer cette coupure au spectre de la variable x (Edep).

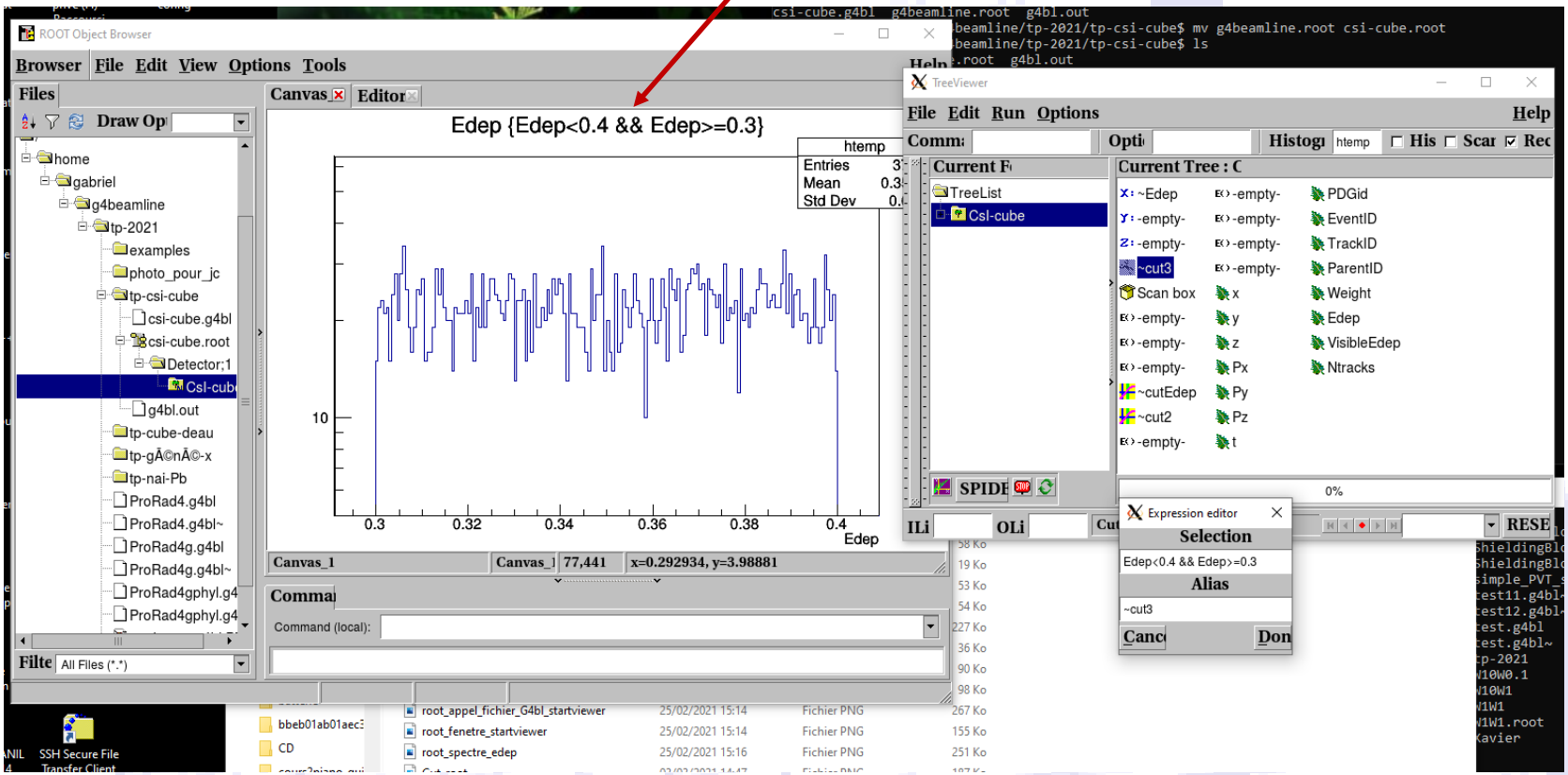
Autres exemples de coupure



Autres exemples de coupure

et

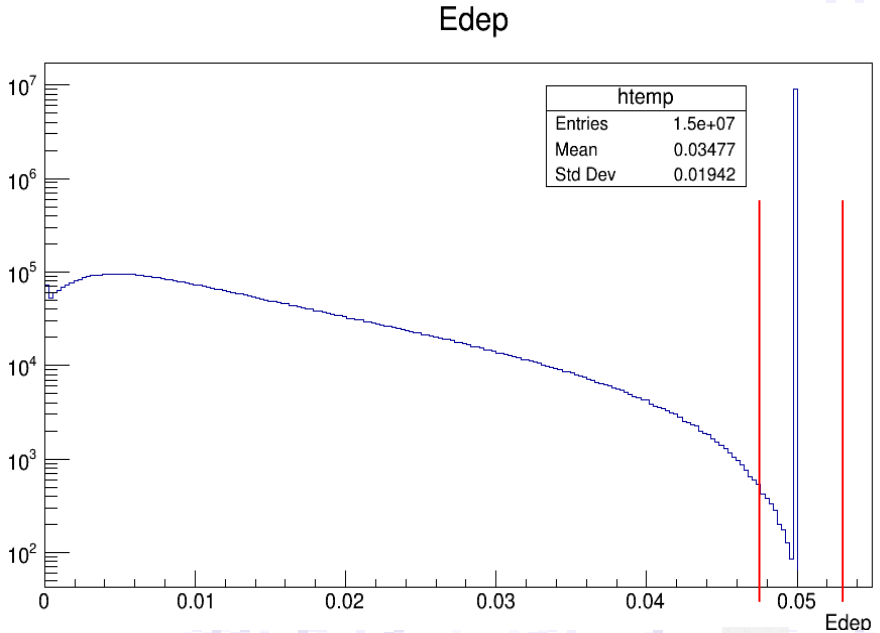
&& : et,
|| : ou.

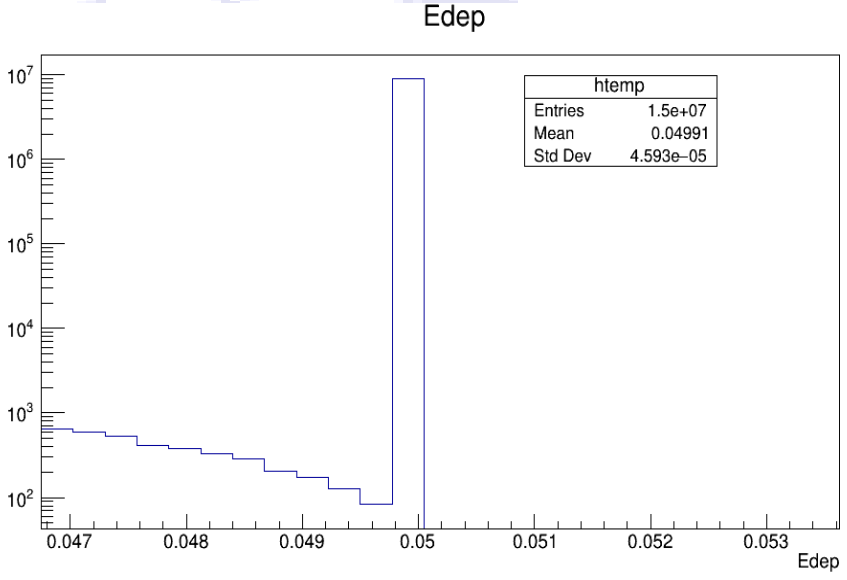


Zoom sur une région du spectre

On peut effectuer un agrandissement d'une région d'intérêt directement sur le spectre.

Sur l'axe x ou y lorsque apparait le symbole « main » faire clic gauche et maintenir le clic jusqu'à la fin de la région du spectre à agrandir





Mouvement clic gauche de gauche à droite

Rmq: On peut revenir à la situation précédente en clic droit et sélectionner unzoom
En plaçant le curseur de la souris sur le pic on peut obtenir son contenu.
On doit voir aussi apparaitre l'information sum qui doit indiquer la somme du spectre visible dans la fenêtre (nombre de coups total).

Visualisation des spectres :

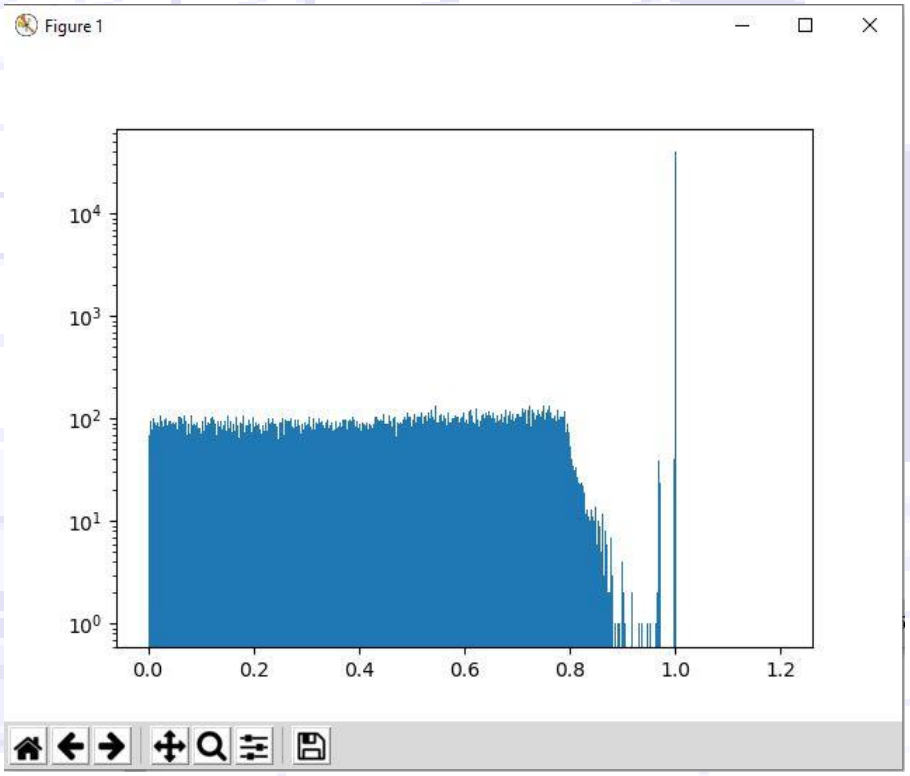
Sous Python:

La visualisation des spectres et l'analyse des spectres .root peut se faire sous python à partir de l'application uproot.

Il faut taper la ligne de commande suivante dans un terminal:
source /opt/python-app/uproot/bin/activate

Executer la commande csi-cube.py qui va lancer la lecture du fichier .root associé au programme et executer les instructions du programme csi-cube:

Exemple pour le programme csi-cube, tracer le spectre de dépôt d'énergie des photons de 1MeV dans un cube Csi:



Visualisation des spectres :

Sous Python: programme d'analyse et de visualization csi-cube.py

```
import numpy as np
import matplotlib.pyplot as plt
import uproot
import awkward as ak
from numpy import log as ln


#tutorial at: https://masonproffitt.github.io/uproot-tutorial/03-trees/index.html

##############################################################
#On importe les fichier root
##############################################################


file_csi = uproot.open('csi-cube.root')          #Importation des fichier
#filePb= uproot.open('Pb1.root')
#fileG=uproot.open('Gant1.root')
#filed2=uproot.open('detecteur2gant.root')

#####################################################################

keys_csi=file_csi.keys()  #Keys prend la forme d'une liste contenant les différents nom des  detecteurs
print(keys_csi)

class_csi=file_csi.classnames() #En affichant class1 on peut voir a quoi correspond chaque
   noms des repertoires ex {'Detector/CsI-cube;1':'TNtuple'} tuple : liste de données qui sont pas modifiable
print(class_csi)
```

Importation des fonctions utiles:

Importation du ou des fichiers .root:

Lecture des données brutes du fichier .root:

```
gabriel@ubuntu22:~/python$ python csi-cube2.py
['Detector;1', 'Detector/CsI-cube;1']
{'Detector;1': 'TDirectory', 'Detector/CsI-cube;1': 'TNtuple'}
['x', 'y', 'z', 'Px', 'Py', 'Pz', 't', 'PDGid', 'EventID', 'TrackID', 'ParentID', 'Weight', 'Edep', 'VisibleEdep', 'Ntracks']
[1, 0.37, 0, 0, 1, 0, 1, 0, 1, 0, ..., 1, 1, 0, 0.675, 0, 0.999, 0, 0, 1, 0]
Traceback (most recent call last):
```

Programme d'analyse et de visualisation csi-cube.py

```
tree_csi = file_csi['Detector/CsI-cube']
tree_csi.keys()   #On importe les différentes données ("énergie,masse,ID....") sous formes de liste.
print(tree_csi.keys())

branches_csi=tree_csi.arrays() #Branche est un tableaux conetenant les données;
       Array[Nbphoton: 2, Phonton_edep: [10.8, ... -1, 1]]
branches_csi['Edep'] #Tableau des énergies
print(branches_csi['Edep']) #Affiches le tableau d'énergoies


#keysPb=filePb.keys()
#classPb=filePb.classnames()
#treePb = filePb['Detector/CdTe-cubePb']
#treePb.keys()
#branchesPb=treePb.arrays()
#branchesPb['Edep']
```

Lecture possible
des données d un autre fichier .root:

```
##########################################################################################################
#Histograms pour une entrée de 1e7
##########################################################
```

Affichage d'un histogramme de données :

```
plt.yscale('log')  #Affiche en échelle logarithmique selon l'axe des y
[y,x,z]=plt.hist(branches_csi['Edep'],bins=500,range=(0,1.2)) # x est le tableau des énergies,
y le nombre de coups et z est un caractère donnant le types des élements du tableau correpondant au fichier file
#[yg,xg,zg]=plt.hist(branchesG['Edep'],bins=500,range=(0, 0.5)) #bins représente le nombre de canaux,
range#=(0,0.5) permet de choisir l'intervalle des valeurs choisis
#[ypb,xpb,zpb]=plt.hist(branchesPb['Edep'],bins=500)
```

Programme d'analyse et de visualisation csi-cube.py

```
#############################################################################
#Exporter les listes sur excel
#pip install openpyxl
#pip install xlwt

#############################################################################¨
import pandas as pd
import xlwt
from xlwt import Workbook

col1 = "Edep"          #Nom de la première colonne case A1
col2 = "Nombre de coups"
data = pd.DataFrame({col1:Lyg,col2:Lxg}) #Data prend la valeur des données voulue dans chaque colonne :
col1 prend les valeurs de la liste Lyg donc le nombre de coups dans le gant).

data.to_excel('sample_data.xlsx', sheet_name='sheet1', index=False) #Création d'un fichier excel du nom sample_data

#######################################
#Affichage des courbes
#######################################

plt.figure(2)   #Nom de la figure
plt.title('Nombre de coup pour une entrée de spectre plat')
plt.xlabel('Edep (MeV)')  #titre de l'axe des abscisses
plt.ylabel('nombre de particules')  #Titre de l'axe des ordonnées
plt.yscale('log')  #Affiche en échelle logarithmique selon l'axe des y
#plt.plot(Lx,Ly, linestyle='-', color='pink',label='Plomb')
plt.hist(branches_csi['Edep'],bins=100,range=(0,2))
plt.legend(loc="lower right") #Position de la légende

plt.show()
```

Transfert possible de données sous un fichier excel :

Repésentation d'un spectre avec légendes:

Généralités sur le programme

Remarque générale: Un code ne peut pas tout résoudre et il est comme une machine à calculer il fait ce qu'on lui demande. Il faut donc un minimum d'esprit critique sur les résultats qui sont générés par le code. En d'autres termes, il faut déjà avoir une petite idée de ce que l'on peut obtenir ou au moins de ce qui n'est pas normal d'obtenir. Important: Conserver toujours une version du programme qui fonctionne avant de modifier le programme. Cela permet de vérifier en cas de dysfonctionnement que l'erreur vient bien du programme qui a été modifié ou du serveur.

Dans les codes de simulations de physique on retrouve les éléments suivants à bien renseigner pour éviter les erreurs de calculs :

Géométrie: Première étape indispensable entrée correctement la géométrie
Forme, dimension, type de matériau, composition, densité……..

Particules: Type de particules étudiées et caractéristiques:
Energies, forme du faisceau (rectiligne , isotrope , dispersion angulaire et en énergie)

Physics lists: Bien s'assurer que les tables de références des modèles utilisées par le code pour la physique que l don veut étudier soit bien adaptées à notre problème : section efficace, collisions, réactions, production des particules et photons,….

Fichier de sortie: Visualisation des interactions dans les détecteurs choisis
Liste des particules et des spectres d'intérêts.
Rmq: nom générique (g4beamline.root). Il sera effacer à la simulation suivante.
Il faut donc modifier le nom si on veut le conserver.

Divers: Nombre de particules tests , temps de simulation,
précision des résultats, réduction de variance..

Structure Fichier g4bl

csI-cube.g4bl

Titre programme et commentaire

ligne non lue

Nommé le fichier de sortie exple ici g4bl_results Sinon par défaut il s'appelera g4beamline.root et sera remplacer par la simulation suivante. (Ne marche pas au bâtiment E).

Physic list
À ne pas modifier à priori

Paramètre
A ne pas modifier à priori

Code couleur des particules

```
*            cube  scintillateur  CsI
*
* Simple example g4beamline input file:
*       a some MeV gamma or e-  beam is sent towards scintillator detect

#####################################################################
#         output file name without .root extension
#####################################################################
param histoFile=g4bl_results
output $histoFile

#####################################################################
#                         Physic list
#####################################################################

physics FTFP_BERT_PEN       # seems best for low energy
#physics QGSP_BERT_PEN
# _PEN, _EMZ, _LIV

#####################################################################
#                       Parameter
#####################################################################

# param epsMin=2.5E-8 // no change
# param minStep=0.001 // no change
# param maxStep=0.1 //slows down a lot

#####################################################################
#####################################################################
#                       Color Code
#  Red=1,0,0    Green =0,1,0    blue=0,0,1    yellow=1,1,0
# opacity=number4 0 to 1
#####################################################################

particlecolor e-=1,0,0 e+=0,0,1 gamma=1,1,0


#####################################################################
```

ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

Définition faisceau
Vers les Z >0

Energie donné en p (MeV/c)
(impulsion)

Faisceau électron
en commentaire

Faisceau gamma
Utilisé ici

100000 événements
Départ en z=0.
Propagation vers Z>0

sigma = dispersion

Définition des matériaux

Mélange et % en masse
density = masse volumique

Définition du détecteur
Avec transparence (opacité)

Titre sur figure de géométrie

```
#########################################################################
#                          BEAM and nEvents
# the beam is nominally headed in the +Z direction, momentum in MeV/c
#
#########################################################################

# Example for e-:
# beam gaussian particle=e- nEvents=10000 beamZ=0.0 sigmaX=1.0 sigmaY=1.0
sigmaXp=0. sigmaYp=0. meanMomentum=sqrt(0.01*(0.01+1.022)) sigmaP=0.0
meanT=0.0 sigmaT=0.0

# Example for gamma:
beam gaussian particle=gamma nEvents=100000 beamZ=0.0 sigmaX=0.1
sigmaY=0.1 sigmaXp=0. sigmaYp=0. meanMomentum=1 sigmaP=0.0 meanT=0.0
sigmaT=0.0


#########################################################################
#                  Material
#########################################################################

#define CsI: material (Cs(132.9) I(126.9) tot=259.8) density crytur 4.51
material CsI Cs,0.5115 I,0.4885 density=4.51
#########################################################################
#                          Detector
#########################################################################

# define the detectors with opacity 20%

detector CsI-cube width=50.0 height=50 length=50.0 color=0,1,0,0.2
material=CsI
place CsI-cube    x=-1    y=-1    z=60

#########################################################################
#                       Title
#########################################################################

# label the visualization
label text="CsI-Cube" -30,-30,40
```

Séances pratiques

Simulation 1 : Prise en main logiciels G4bl et Root avec exemple étude Cube-CsI.

Simulation 2 : Spectroscopie gamma avec un scintillateur NaI.

Simulation 3 : Etude d'atténuation des photons par la matière.

Simulation 4 : Etude de l'efficacité d'un détecteur de photons.

Simulation 5 : Dépôt d'énergie des photons et particules chargées dans l'eau.

Simulation 6 : Méthode d'identification de particules chargées E ΔE. (Point de rebroussement).

Simulation 7 : Etude de la fluorescence X**.**

Simulation 8 : Mise en évidence du phénomène d'accumulation dit aussi de build-up.

Simulation ++ : Si le temps le permet un perfectionnement sera proposé permettant d'améliorer les simulations et les connaissances dans l'analyse des données par root.